U1

- [ ] Network edge pg 20 - [ ] Numericals) pg 31, 32, 45 - 48, 129 - 146, U2) 263 - 266

Terms

- Computer Network - hosts interconnected through communication links and packet switches
- Internet - computer network that interconnects wider devices around the world
- ISP - business entity which provides internet access to end-systems
- POP (Point of Presence) - place where end systems connect to ISP
- Network Edge #todo
- Network Core
- Types of switching

```
* Circuit - resource allocation before data transfer
    * TDM (Time Division Multiplexing)
    * FDM (Frequency division multiplexing)
* Packet - Multiplexing of packets
```

- Access Network Nomenclature

```
* Size - LAN, WAN, Home networks
* Topology - Tree, Star, Ring, bus, point-to-point
* Physical media - Wired (DSL, Cable, Fiber to home) or wireless
```

- Delays

```
* Processing Delay - time taken to inspect a packet
* Queuing delay - time spent by paket in queue before processing
* Transmission delay - time taken to push a packet on to the link
* Packet Switch delay - sum of all above delays
* Propogation delay - time for packet to travel over link
* End-to-end delay - total time spent from source to destination
```

- Traffic Intensity - La/R (where L - packet size, R - link rate, a - arrival rate)
- Throughput - rate at which destination host recieves the packets
- Encapsulation - After performing a sub-task, the source appends some information to the packet called header to the data it processed
- Decapsulation - Before performing a sub-task the destination removes the corresponding header

Protocol layers

- Requirements - syntax, Semantics, Timing
- Types

```
   * TCP/IP
       * From ARPANET
       * 5 Layers
           * Application
               * generate/recieve data
               * data is called messages
               * messages can be big in size
               * can have QoS requirements
           * Transport
               * responsible for the QoS
               * does multiplexing (sender), demultiplexing (reciever)
               * Maps each message to corresponding process
               * Appends a header to each message
               * segment = message + header
           * Network
               * fragment segments into packets
               * moves packets hop by hop based on src & dest. IP addr
ess
               * appends new header to each packet
               * datagram = packet + header
           * Link
               * pushes packet into link
               * can forward frames using mac address
               * appends new header to each packet
               * frame = packet + header
               * check for error in frame
               * provide synchronization in reciever
           * Physical
               * physical interface
               * converts binary to signals
               * does modulation, demodulation, transmission, receptio
n and filtering of signals
   * OSI
```

Application architecture

- Client - Server

```
   * Client initiates, server responds to requests
   * Server can handle concurrent connections
```

- Peer to Peer

```
   * any host can recieve / send data
   * hosts allocate resources to help each other
   * P2P architectures are self scalable
   * distributed algorithm used to
       * maintain state information
       * file sharing
```

Transport layer services

| Application | App. Layer Protocol | Transport Protocol | | ---------- | ---------- | ---------- | | E-mail | SMTP | TCP | | Remote terminal access | Telnet | TCP | | Web | HTTP | TCP | | File transfer | FTP | TCP | | Streaming media | HTTP | TCP | | Internel telephony | SIP, RTP or proprietary | UDP or TCP |

Web and HTTP

- Client (Web browser) sends HTTP request specifying object requested
- Web server sends HTTP response message which may contain requested object
- Types

```
   * Persistent
       * In persistent HTTP connections, only 1 TCP connection is esta
blished and all objects are fetched back to back
       * saves total access time
       * TCP buffers must be allocated for each of connections
       * server closes connection after specific period of inactivity
       * Used in HTTP/1.1 (upto 6 parallel TCP connecitons)
       * Used in HTTP/2 (multiplexing, message prioritization, server
pushing)
   * Non-persistent
       * Seperate TCP connection to fetch each object
       * Assume neglible size for HTTP request message
       * Total acces delay / object = transmission delay + 2 x RTT
       * Socket number of web server in 80
```

HTTP Message format

- Request message

```
   * Entity body empty (download) or non-empty (upload)
```

- GET message
- Response
- Cookies - meant for access through user identification

Web Caching

- Web cache is a network entity that satisfies HTTP requests on behalf of an origin web server
- Stores the copies of recently requested objects
- It reduces infrastucture cost and access delay in large organisations
- Content Delivery Networks (CDN) give web caching too

Email

- Mailbox <-> user agent --uses--> Uses POP3, IMAP, HTTP
- Outgoing messages in mailbox --uses--> SMTP (Simple Mail Transfer Protocol)

```
   * SMTP - pushes, recipient - pulls
   * Socket number 25
```

- Messages between mail servers are encoded in ASCII

| HTTP | SMTP | | ---------- | ---------- | | Allows different encodings | Only ASCII encoding supported | | Pull type protocol | Push type protocol | | Distinguishes b/w object types | Doesn't distinguish b/w object types |

DNS

- Domain Name System
- A way to resolve the host names into IP addresses
- DNS servers

```
   * Unix machines running Berkeley Internet Name Domain (BIND) softwa
re
   * Types
       * Root DNS servers
           * 1st level of DNS servers contacted by clients to query DN
S
           * 13 root servers accross the world
       * TLD DNS servers
           * maintain domain level information
       * Authoritative DNS servers
           * maintain various DNS records corresponding to registered
hosts
       * Local DNS servers
           * query the DNS hierarchy on behalf of clients
```

- Uses UDP for transport layer protocol
- port 58 of DNS server
- Query mechanisms

```
   * Iterative
   * Recursive
```

| Type | Name | Value | | -------- | ---------- | ---------- | | A | Hostname | IP Address | | NS | Hostname | Name od Authoritative DNS | | CNAME | Alias host name (web server) | Canonical name of web server | | MX | Alias host name (mail server) | Canonical mail server name |

Process

1) Send HTTP request to webserver for the 1st time 2) DNS server performs decapsulation and reads the DNS query. 3) Then, it generates a DNS reply having the IP address of the web server 4) DNS server encapsulates the reply in a UDP segment and passes it 5) Upon receiving the DNS reply, the encapsulation of the TCP handshake (i.e., TCP connection request) segment resumes using the IP address obtained for the web server. 6) This TCP segment is passed to the web server 7) Web server replies with a TCP handshake (i.e., TCP

connection grant) of its own from port 80 8) Upon receiving the TCP handshake from the web server, the host performs encapsulation of the HTTP request and then sends it to the web server

Hierarchy of DNS servers

- Root DNS servers

```
  * com DNS servers
      * facebook.com DNS servers
      * amazon.com DNS servers
  * org DNS servers
      * pbs.org DNS servers
  * edu DNS servers
      * nyu.edu DNS servers
      * umass.edu DNS servers
```

DNS caching and vulneribilities

- Caching

```
  * Reduce network traffic
  * Reduce delay in DNS response
```

- Vulnerabilities

```
  * Denial of service attack
      * DNS server choked with queries
  * Spoofing
      * Client is choked with responses
  * Man-in-the-middle
      * Message is altered
```

Video Streaming and CDN

- Problem

```
  * Bandwidth bottleneck
  * Video avaiability at client desired bit rate
```

- Solutions

```
  * DASH (Dynamic Adaptive Streaming over HTTP)
      * varying video resolution is response to changes in available
bandwidth
      * Multiple version with varying bit rate made available at cont
ent server
      * A client opens a TCP connection to content server
      * Server sends back manifest file
      * Manifest file contains URL (designated for each version of co
ntent) and it's bit rate
      * client's application buffers the the reciever chunks before p
lay out
      * objective is to ensure quality of experience is chosed by cli
```

```
ent
   * CDN (Content Distribution Networks)
       * objective - maintain videos closer to clients and resolve ser
ver assignment
       * distributed architecture of server clusters to store content
```

| Enter deep | Bring home | | ---------- | ---------- | | Large number of small clusters | Small number of large clusters | | Deployed in ISPs | Deployed in IXPs | | Challenge :- Maintainence, overhead | Challenge :- delay and throughput |

Transport Layer

- Provides logical connection between process
- Protocols are implemented in the end systems not network routers
- Performs multiplexing and demultiplexing of segments in a host
- Performs error detection and in-order assembly of segments
- Can provide QoS support for applications
- Protocols

```
   * UDP
       * Connectionless protocol
       * No acknowledgment of transmitted segments
       * No throughput regulation
       * No service guarantees
       * Small packet header overhead (8 bytes)
       * Suited for application which are time sensitive
       * Segments - src port, dest. port, length, checksum
       * Checksum -
           * Sender -
               * Split the segment into 16-bit numbers and sum them
               * Wrap around carry (if any)
               * Take 1’s complement of the sum (call this UDP checksu
m)
           * Reciever -
               * Recompute checksum including UDP checksum
               * If answer is all 1s then it means no error has occurr
ed
   * TCP
       * Connection oriented
       * Adaptive throughput based on network congestion
       * Supports flow control at recieving node
       * Guarantees reliable data transfer
```

- Sockets

```
   * Each active application is associated by one or more sockets assi
gned by the operating system
   * Used in multiplexing and demultiplexing of segments
```

- Multiplexing - Segments leaving different sockets in a end-system are interleaved so that the network layer can assign the source IP address
- Demultiplexing - Segments arriving from the network layer with the same destination IP address, corresponding to the end-system, are separated and delivered to respective sockets